Hardware e acesso aos dispositivos

www.4linux.com.br

Sumário

Índice de tabelas

Índice de Figuras

Capítulo 1

Hardware e Dispositivos

1.1. Objetivos

- Iremos aprender de que forma os dispositivos de hardware são mapeados e manipulados no GNU/Linux;
- Para que esse assunto faça mais sentido, primeiramente veremos alguns conceitos sobre arquitetura de computadores e dispositivos de hardware.

1.2. Mãos a obra

Vamos estudar Arquitetura de Computadores e Dispositivos de Hardware

Podemos dividir um computador em 3 partes principais:

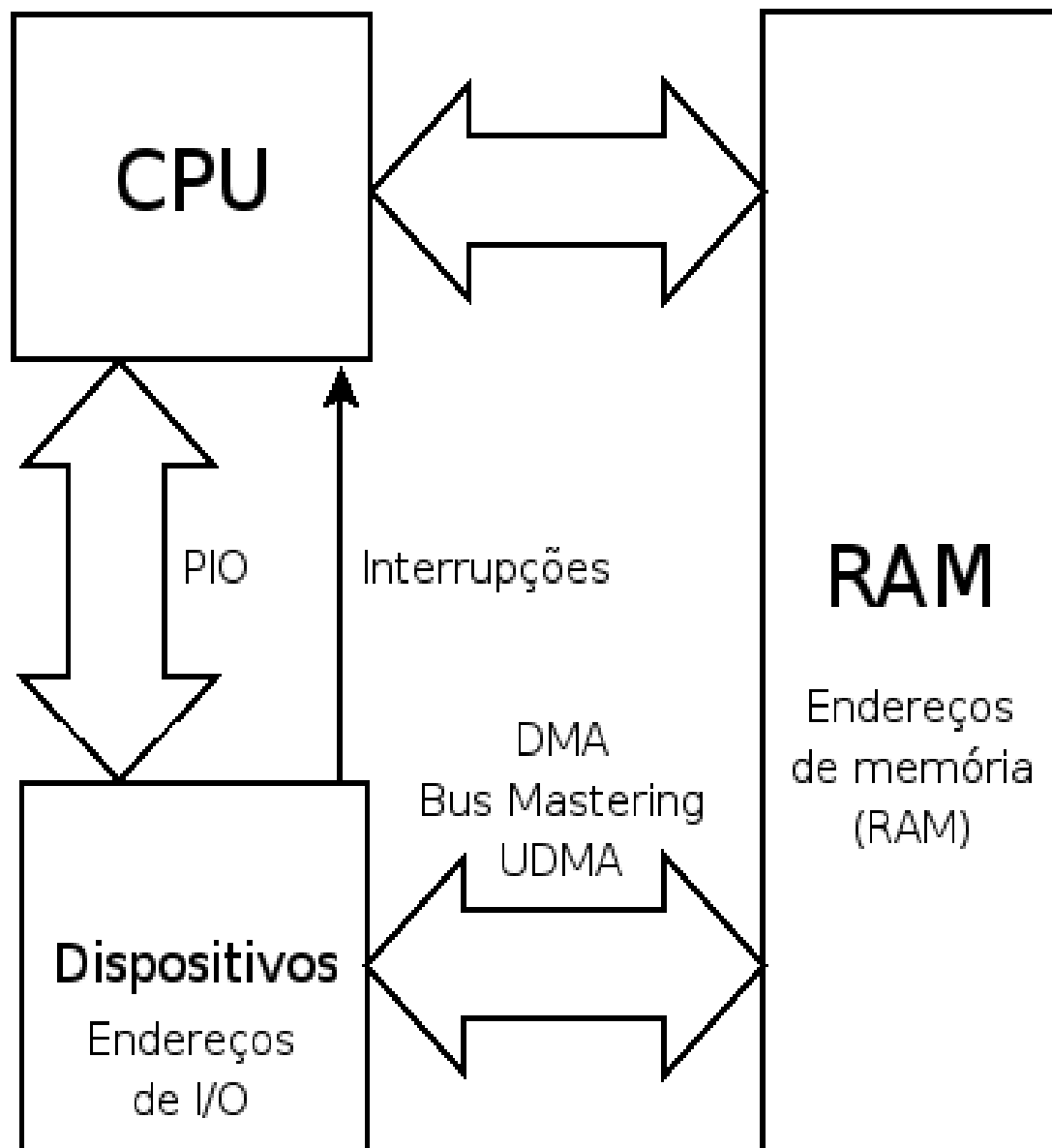
- CPU;
- Memória RAM;
- Dispositivos.

A "CPU", muitas vezes denominada como o cérebro do computador, é responsável por executar todo o processamento das informações, que são armazenadas na memória "RAM".

Mas, um computador não tem muita utilidade se não for capaz de se comunicar com o mundo exterior. Um teclado e um monitor, ou uma rede, são exemplos de meios de comunicação. Até mesmo um simples botão (no lugar do teclado) e uma lâmpada (no lugar do monitor) poderiam ser considerados como exemplo. A esses elementos damos o nome de dispositivos de "hardware", e incluem interfaces de rede, controladoras de disco, as próprias unidades de disco, portas seriais, paralelas e USB, apenas para exemplificar.

Arquitetura do computador é o nome que damos à forma como essas 3 coisas são organizadas numa máquina.

A figura a seguir ilustra a arquitetura típica dos "Pcs" →

Ilustração 1: Arquitetura

Capítulo 2

Acesso aos dispositivos

2.1. Objetivos

- Entender como é feito o acesso aos dispositivos;
- Entender como são conhecidos alguns dispositivos.

2.2. Mãos a obra

O acesso aos dados da memória “RAM” é feito de forma rápida e eficiente através de otimizados canais de comunicação. Entretanto, o acesso aos dispositivos é mais lento, e as tecnologias responsáveis por essa função podem ser dividas em duas categorias.

A primeira, chamada “PIO - Programmed Input/Output”, envolve a “CPU” na transferência das informações. Para identificar os dispositivos, são associados a eles os chamados endereços de “I/O - Input/Output”. Assim, por exemplo, a “COM1” tem o endereço “3F8h”, a “LPT1” o endereço “378h”. Na verdade, um certo intervalo desses endereços são utilizados para cada dispositivo. Esses endereços podem ser consultados no arquivo “/proc/ioports”.

Além desses endereços, em alguns casos temos um interrupção associada a um dispositivo. Isso porque, como são mais lentos que a “CPU”, precisam de algum mecanismo para informar à “CPU” de que o trabalho terminou. Do contrário, a “CPU” teria de ficar constantemente consultando o dispositivo para saber quando enviar ou ler o próximo “byte”, e consequentemente perdendo tempo.

A cada dispositivo, é associada uma interrupção. Entretanto, o número disponível de interrupções é limitado, e por essa razão, pode faltar alguma e/ou ocorrer os famosos “conflitos de interrupção”. As interrupções utilizadas podem ser consultadas no arquivo “/proc/interrupts”.

Entretanto, a tecnologia “PIO” limita a velocidade de transferência de dados. Ela é apropriada apenas para dispositivos como teclado, portas seriais e paralelas, unidades antigas de CD-ROM, etc.

Outro problema relacionado a ela é o envolvimento da “CPU”. Isso porque, vários ciclos de processamento são perdidos no processo de transferência dos dados, o que se agrava tanto quanto maior for a velocidade dessa transferência.

Para contornar essa situação, foi criado o “DMA Direct Memory Access”. Essa tecnologia permite que o dispositivo acesse diretamente a memória “RAM”,

escrevendo ou lendo dados, sem interferência da “CPU”. Para isso, são utilizados os chamados “canais de DMA”, um para cada dispositivo e também uma controladora de “DMA”. Os canais utilizados podem ser consultados no arquivo “/proc/dma”.

Mas essa tecnologia, desenvolvida para os antigos barramentos “ISA”, também ficou ultrapassada, e cedeu lugar ao “Bus Mastering”. Nesse caso, o próprio dispositivo faz todo o controle de acesso a memória “RAM”, de modo que os canais de “DMA” não são mais necessários. Essa nova tecnologia permitiu o surgimento do “UDMA - “Ultra DMA”.

Alguns dispositivos legados possuem endereços e interrupções padrões.

Dispositivos	Nome no Linux	End. Hex	Int.
COM1	/dev/ttyS0	3 F 8	4
COM2	/dev/ttyS1	2 F 8	3
COM3	/dev/ttyS2	3,00E+008	4
COM4	/dev/ttyS3	2,00E+008	3
LPT1	/dev/lp0	378	7
LPT2	/dev/lp1	278	5

Capítulo 3

Gerenciando

3.1. Objetivos

- Visualizar alguns arquivos;
- Entender como funciona os dispositivos.

3.2. Troubleshooting

Você já reparou na saída do comando "df"?

Quando trabalhamos com Linux, esquecemos de alguns detalhes, repare na saída do comando:

```
# df -h
```

Repare na saída do comando:

```
udev                   10M  112K  9,9M   2% /dev
```

Então, como podemos ver, não é porque o /dev está dentro de "/" que utiliza "ext3" que ele vai ser ext3. O /dev armazena os dispositivos do sistema, seu sistemas de arquivos é o "udev". Uma das diferenças que existem do antigo "DEVFS" para o UDEV que estamos utilizando está na criação de dispositivos. No DEVFS eram criados todos os dispositivos possíveis e imagináveis. Começou a ser utilizado o UDEV a partir da série do kernel 2.6.12.

Veja o kernel que estamos utilizando:

```
# uname -a
```

Veja somente a versão:

```
# uname -r
```

Para visualizarmos Kernel e Distro:

```
#  cat  /proc/version
```

Para visualizarmos as partições criadas na máquina:

```
#  cat  /proc/partitions
```

Para ver as partições montadas:

```
#  cat  /proc/mounts
```

Endereços de IO:

```
#  cat  /proc/ioports
```

Modelo de processador:

```
#  cat  /proc/cpuinfo
```

Um desafio, gostaria que você descubra o que é "bogomips"!!!

Conhecer os sistemas de arquivos, dispositivos, kernel, são fundamentais para trabalhar com Linux.

Para visualizar os sistemas de arquivos:

```
#  cat  /proc/filesystems
```